

AD-A074 381

NAVAL POSTGRADUATE SCHOOL MONTEREY CA

F/G 17/2

A DECENTRALIZED ALGORITHM FOR FINDING THE SHORTEST PATHS IN DEF--ETC(U)

JUL 79 J Y YEN

UNCLASSIFIED

NPS55-79-015

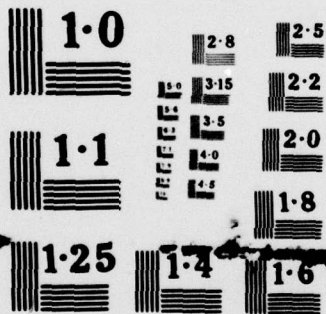
NL

| OF |  
AD-  
A074381



END  
DATE  
FILMED

11-79  
DDC



NATIONAL BUREAU OF STANDARDS  
MICROCOPY RESOLUTION TEST CHART

LEVEL II

25

NPS55-79-015

NAVAL POSTGRADUATE SCHOOL  
Monterey, California

ADA074381



DDC  
SEP 27 1979  
A

DDC FILE COPY

A DECENTRALIZED ALGORITHM FOR FINDING  
THE SHORTEST PATHS IN  
DEFENSE COMMUNICATIONS NETWORKS

by

Jin Y. Yen

July 1979

Approved for public release; distribution unlimited.

79 09 24 129

NAVAL POSTGRADUATE SCHOOL  
MONTEREY, CALIFORNIA

Rear Admiral T. F. Dedman  
Superintendent

Jack R. Borsting  
Provost

The work reported herein was supported by the Foundation Research Program of the Naval Postgraduate School with funds provided by the Chief of Naval Research.

Reproduction of all or part of this report is authorized.

This report was prepared by:

*Jin Y. Yen*

Jin Y. Yen, Adjunct Professor  
Department of Operations Research

Reviewed by:

*Michael G. Sovereign*  
Michael G. Sovereign, Chairman  
Department of Operations Research

Released by:

*William M. Tolles*  
William M. Tolles  
Dean of Research

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER 14 NPS55-79-015	2. GOVT ACCESSION NO.	3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle) 6 A Decentralized Algorithm for Finding the Shortest Paths in Defense Communications Networks		5. TYPE OF REPORT & PERIOD COVERED 9 Technical rept.
7. AUTHOR(s) 10 Jin Y. Yen		8. CONTRACT OR GRANT NUMBER(s)
9. PERFORMING ORGANIZATION NAME AND ADDRESS Naval Postgraduate School Monterey, Ca 93940		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS 16 61152N; RR 000 01-01 N0001479WR90027
11. CONTROLLING OFFICE NAME AND ADDRESS Naval Postgraduate School Monterey, Ca 93940		12. REPORT DATE 11 Jul 79
14. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office) 12 17p.		15. SECURITY CLASS. (of this report) Unclassified
16. DISTRIBUTION STATEMENT (of this Report) Approved for public release; distribution unlimited.		
17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report) 17 RR 00000101		
18. SUPPLEMENTARY NOTES		
19. KEY WORDS (Continue on reverse side if necessary and identify by block number) Network Shortest paths Decentralized algorithm		
20. ABSTRACT (Continue on reverse side if necessary and identify by block number) This paper presents a decentralized shortest path algorithm which finds the shortest distances between all pairs of nodes without requiring that any particular node have information about the complete topology of the network. The algorithm requires at most $N^{3/2}$ additions, $N^{3/2}$ comparisons, and $N^{3/2}$ transmissions of simple messages between individual nodes. The computational upper bound of the present algorithm is lower than that of Dijkstra's centralized shortest path algorithm and is $1/N$ of the upper bound of Abram and Rhodes' decentralized shortest path algorithm.		

DD FORM 1 JAN 73 1473

EDITION OF 1 NOV 65 IS OBSOLETE  
S/N 0102-014-6601

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

251450

Jue



A DECENTRALIZED ALGORITHM FOR FINDING THE SHORTEST PATHS  
IN DEFENSE COMMUNICATIONS NETWORKS

by

Jin Y. Yen  
Naval Postgraduate School  
Monterey, CA 93940

ABSTRACT

This paper presents a decentralized shortest path algorithm which finds the shortest distances between all pairs of nodes without requiring that any particular node have information about the complete topology of the network. The algorithm requires at most  $\frac{1}{2}N^3$  additions,  $\frac{1}{2}N^3$  comparisons, and  $\frac{1}{2}N^3$  transmissions of simple messages between individual nodes. The computational upper bound of the present algorithm is lower than that of Dijkstra's centralized shortest path algorithm and is  $1/N$  of the upper bound of Abram and Rhodes' decentralized shortest path algorithm.

Accession For	
NTIS GRA&I	<input checked="checked" type="checkbox"/>
DDC TAB	<input type="checkbox"/>
Unannounced	<input type="checkbox"/>
Justification	
By _____	
Distribution/ _____	
Availability Codes	
Dist	Avail and/or special
A	

A DECENTRALIZED ALGORITHM FOR FINDING THE SHORTEST PATHS  
IN DEFENSE COMMUNICATIONS NETWORKS

by

Jin Y. Yen

The problem of finding shortest paths has a wide variety of applications in communication networks [9], [12], [13]. Many authors, including Dantzig [3], [4], Ford and Fulkerson [8], Bellman [2], Dijkstra [5], and Floyd [7], have introduced efficient algorithms for finding the shortest paths in networks. However, in order to apply these algorithms in a communication network it is necessary to establish a central node to gather information concerning the complete network topology so that the algorithms can be executed.

The shortest path algorithms that must be executed at a central node are called centralized shortest path algorithms. The centralized shortest path algorithms have very good computational efficiency. However, they have many disadvantages when applied to military and intelligence communication networks. The major disadvantages of the centralized shortest path algorithms are as follows:

- 1) They make the network more vulnerable. The centralized shortest path algorithms require a central node to execute the algorithm. When the central node or the links directly connected to it are destroyed, the network completely loses its ability to function.
- 2) They make it more difficult to maintain the security of the network. The central node has complete information concerning the whole network. One has to penetrate

or monitor only the central node to obtain information concerning the whole network.

- 3) They require substantial effort to transmit information between the central node and other nodes in the network. The centralized shortest path algorithms require that individual nodes inform the central node of how they are linked to other nodes in the network and what the associated link lengths are. These algorithms also require that the central node disseminate the resulting solution to individual nodes. Substantial effort is necessary to transmit this information between the central node and all other nodes in the network.

Due to the disadvantages of centralized shortest path algorithms, it is desirable to develop another type of algorithm that does not depend on the existence of a central node. To find the shortest paths from all nodes to a destination node, the new type of algorithm requires that each individual node communicate only with its adjacent nodes with such simple information as what is its current shortest distance to the destination node. Since this type of algorithm does not depend on the existence of a central node and solves the problem locally using only local information, it is called a decentralized shortest path algorithm.

While there are numerous centralized shortest path algorithms in the literature [6], [11], [14], there is only one published



decentralized shortest path algorithm known to the author. In reference 1, Abram and Rhodes present a decentralized shortest path algorithm based on the principle of a centralized shortest path algorithm of Ford and Fulkerson [8]. To apply the algorithm of Abram and Rhodes, the individual nodes communicate only with their adjacent nodes with simple information. However, due to the fact that this algorithm does not assume knowledge of the complete network topology, it requires more repetitive computations than the original Ford and Fulkerson algorithm; these repetitive computations produce a substantial increase in the computation bound of the algorithm. To find the shortest distances between all pairs of nodes in an  $N$ -node network, the Abram and Rhodes algorithm can require up to  $\frac{1}{2}N^4$  additions and  $\frac{1}{2}N^4$  comparisons which are approximately  $N$  times higher than that of Dijkstra's [5], Floyd's [7], and Dantzig's [4] algorithms. Also, in order to carry out the  $\frac{1}{2}N^4$  computational steps, the algorithm requires individual nodes to communicate with their adjacent nodes up to  $\frac{1}{2}N^4$  times, which appears to be more overburdening than the required  $\frac{1}{2}N^4$  computational steps.

The purpose of this paper is to present a new decentralized shortest path algorithm for finding the shortest distances between all pairs of nodes in an  $N$ -node directed network using at most  $\frac{1}{2}N^3$  additions,  $\frac{1}{2}N^3$  comparisons, and  $\frac{1}{2}N^3$  transmissions of simple messages between all nodes in the network. The necessary assumptions for applying the new algorithm are as follows:

- 1) Each node in the network is equipped with transmission and computation facilities and a timing device called a clock.
- 2) Each node  $J$  knows a set of nodes, called FROM nodes, each of which is connected to node  $J$  by a directed link leading from the FROM node to node  $J$ .
- 3) Each node  $J$  knows a set of nodes, called TO nodes, each of which is connected to node  $J$  by a directed link leading from node  $J$  to the TO node. Each node  $J$  also knows the lengths of the links connecting node  $J$  to the TO nodes.

In an  $N$ -node directed network, let

$I, J, K, L = 1, 2, \dots, N$ , be the nodes of the network,

$(I, J)$  be the link connecting node  $I$  to node  $J$ ,

$D(I, J) \geq 0$  be the length of link  $(I, J)$ ,

$F(I, J)$  be the distance of the tentative shortest path from node  $I$  to node  $J$ . Initially, all  $F(I, J)$ 's are set to  $\infty$ ,

$T[F(I, J)]$  be the finite length of time defined to represent the corresponding value of  $F(I, J)$ . Initially, all  $T[F(I, J)]$ 's are set to  $\infty$ ,

$C$  be a constant such that  $C = F(I, J)/T[F(I, J)]$ .

The new algorithm for finding the shortest paths from all nodes to a destination node K is as follows. In order to simplify the description of the algorithm we assume without loss of generality that no time is necessary to transmit, to receive, and to process the information. Of course, we assume all clocks are synchronized.

(Algorithm)

Step 1. At time 0, the destination node K sends each of its adjacent FROM nodes J a simple message: "K".

Step 2. After receiving the message, each of node J does the following:

A. Label the node that has just sent the message node L and delete node L from its own list of FROM nodes.

B. Read the clock and let  $T[F(L,K)]$  equal the time it reads from the clock and let  $F(L,K) = C \cdot T[F(L,K)]$ .

C. Update  $F(J,K)$  by

$$F(J,K) = \min[F(J,K), D(J,L) + F(L,K)].$$

D. Let  $T[F(J,K)] = \frac{1}{C} \cdot F(J,K)$ .

E. At time  $T[F(J,K)]$ , node J sends its own adjacent FROM nodes a message: "J".

Step 3. Repeat Step 2 until time  $t^*$ , where  $t^*$  is a predetermined constant larger than any  $T[F(J,K)]$ .



At termination of the algorithm, each node J has the following solution to the shortest path to destination node K:

- 1) The distance of the optimal shortest path from node J to the destination node K, which is represented by  $F(J,K)$ , and
- 2) The identity of the second node on the shortest path from node J to node K, which is indicated by the node from which the final  $F(J,K)$  is obtained.

It is clear that some minor changes can be made in the algorithm to compensate for the time lags due to transmission and processing of information. Also, the algorithm can be modified so that node J can send  $F(J,K)$  to its adjacent FROM nodes at time  $T[F(J,K)]$  to save these adjacent nodes from reading their clocks to determine  $T[F(J,K)]$  and  $F(J,K)$ . The present algorithm can be repeated or applied simultaneously to obtain the shortest distances from nodes J to other destination nodes. Of course, when the algorithm is applied to find simultaneously the shortest distances to many destination nodes, additional information identifying the destination nodes must be sent along with such messages as "K" and "J" in Step 1 and Step 2.E. of the algorithm in order to assure proper functioning of the algorithm.

We will now show that the algorithm determines a set of optimal shortest distances from nodes J to the destination node K. At time  $t$ , the set of  $F(J,K)$ 's for those  $T[F(J,K)] > t$ , are the tentative shortest distances from nodes J to destination node K



using the best paths available up to that time. As time passes, the smallest of these tentative  $F(J,K)$ 's, say  $F(J^*,K)$  becomes permanently labeled because at time  $t = T[F(J^*,K)]$  it becomes apparent that there is no other path from node  $J^*$  to node  $K$  that has shorter distances than  $F(J^*,K)$ . On one hand, the tentative  $F(J,K)$ 's becomes permanently labeled as time passes; and, on the other hand, whenever a  $F(J,K)$  becomes permanently labeled it is used to update other tentative  $F(J,K)$ 's. Therefore, at the termination of the algorithm, the  $F(J,K)$ 's thus obtained are the distances of the optimal shortest distances from nodes  $J$  to the destination node  $K$ .

In a connected network, the distances of all shortest paths,  $F(J,K)$ 's, are finite; thus, the times in which they are determined by the algorithm,  $T[F(J,K)]$ 's, are also finite. Therefore the present algorithm determines all permanent shortest distances to node  $K$  in a finite time. As a matter of fact, all permanent shortest distances,  $F(J,K)$ 's, are determined at time  $t = T[F(J^{**},K)]$  where  $F(J^{**},K)$  is the largest of all permanent  $F(J,K)$ 's.

However, the algorithm is not able to detect this fact in order to terminate the algorithm as soon as the last  $F(J,K)$  becomes permanent. Instead, the algorithm terminates at a preset time  $t^*$  where  $t^*$  is larger than any  $T[F(J,K)]$ .

The computational efficiency of the present algorithm appears to be quite good. The computational advantages of the algorithm include:

- 1) The computational effort of the algorithm is proportional to the number of links in the network. Therefore, unlike algorithms such as Dijkstra's [5], the present algorithm requires fewer computations in sparse networks where there are fewer links.
- 2) Unlike Dijkstra's algorithm, the present algorithm does not have to scan for the minimum of all tentative shortest distances in order to sort out the permanent shortest distance; consequently, it saves computations.
- 3) Unlike Ford and Fulkerson's [8], Moore's [10], Bellman's [2], Floyd's [7], and Abram and Rhodes' [1] algorithms, the present algorithm does not use a shortest distance  $F(L,K)$  to update other tentative shortest distances  $F(J,K)$  unless  $F(L,K)$  itself is permanent.

To determine the shortest distances from all nodes  $J$  to a destination node  $K$  in an  $N$ -node complete network the present algorithm requires at most  $(N-1) + (N-2) + \dots + 1 = \frac{1}{2}N^2$  additions and the same number of comparisons to execute the  $N-1$  iterations of Step 2.C. of the algorithm. The algorithm also requires at most  $\frac{1}{2}N^2$  transmissions of such simple message as " $K$ " and " $J$ " in Step 1 and Step 2.E. of the algorithm. As compared with Abram and Rhodes' algorithm [1], the present algorithm has an upper bound equal to only  $1/N$  of the upper bound of their algorithm.

While Dijkstra's algorithm [5] which requires up to  $\frac{1}{2}N^2$  additions and  $N^2$  comparisons to determine all shortest distances to a single destination in an N-node complete network and is believed to be computationally most efficient [6], it is interesting to note that the present algorithm requires only  $\frac{1}{2}N^2$  additions and  $\frac{1}{2}N^2$  comparisons to solve the problem.

#### Acknowledgments

The author would like to thank Professors Michael G. Sovereign, Jack Wozencraft and Donald P. Gaver for their helpful discussions and encouragement of this research.



## REFERENCES

1. Abram, J. M. and I. B. Rhodes, "A decentralized shortest path algorithm," to appear in Proc. of the 16th Annual Allerton Conference on Communication, Control and Computing, University of Illinois, Oct. 4-6, 1978.
2. Bellman, R. E., "On a routing problem," Quart. Appl. Math., Vol. 16, No. 1 (1958), pp. 87-90.
3. Dantzig, G. B., "On the shortest route through a network," Management Science, Vol. 6, No. 2 (1960), pp. 187-190.
4. ———, "All shortest routes in a graph," (Ed.) P. Rosenstiehl, Theory of Graphs, pp. 91-92, Gordon and Breach (1967).
5. Dijkstra, E. W., "A note on two problems in connection with graphs," Numerische Mathematik, Vol. 1 (1959), pp. 269-271.
6. Dreyfus, S. E., "An appraisal of some shortest path algorithms," Operations Research, Vol. 17 (1969), pp. 395-412.
7. Floyd, R. W., "Algorithm 97, shortest path," Comm. ACM, Vol. 5 (1962), p. 345.
8. Ford, L. R., Jr. and D. R. Fulkerson, Flows in Networks, Princeton University Press, Princeton, N. J. (1962), pp. 130-133.
9. Frank, H. and I. T. Frisch, Communication, Transmission and Transportation Networks, Addison-Wesley, Reading, Mass. (1971).
10. Moore, E. F., "The shortest path through a maze," Proc. Int. Symp. on the Theory of Switching, Part II, April 2-5, 1957, The Annals of the Computation Laboratory of Harvard University, 30, Harvard University Press (1959).
11. Pierce, A. R., "Bibliography on algorithms for shortest path, shortest spanning tree, and related circuit routing problems (1956-1974)," Networks, Vol. 5 (1975), pp. 129-149.
12. Schwartz, M., Computer-Communication Network Design and Analysis, Prentice-Hall, Englewood Cliffs, N. J. (1977).
13. Tactical Communications Vulnerability Assessment Model, Vol. II, Book 1 (draft), TTO-ORT-032-75-V2.1, Joint Tactical Communications Office, Fort Monmouth, N. J. (March 1975).
14. Yen, J. Y., Shortest Path Network Problems, Verlag Anton Hain, Meisenheim Am Glan, W. Germany (1975).



# DISTRIBUTION LIST

	No. of copies
Defense Documentation Center Cameron Station Alexandria, VA 22314	2
Technical Information Division Naval Research Laboratory Washington, D.C. 20375	1
Library, Code 0212 Naval Postgraduate School Monterey, CA 93940	2
Library, Code 55 Naval Postgraduate School Monterey, CA 93940	1
Navy Library National Space Technology Lab Attn: Navy Librarian Bay St. Louis, MO 39522	1
Naval Electronic Systems Command Navelex 320 National Center No. 1 Arlington, VA 20360	1
Director Naval Research Lab Attn: Library (ONRL) Code 2029 Washington, D.C. 20375	1
Office of Naval Research San Francisco Area Office 760 Market Street San Francisco, CA 94102	1
Technical Library Naval Ordnance Station Indian Head, MD 20640	1

No. of Copies

Dr. J. Chandra  
U. S. Army Research  
P.O. Box 12211  
Research Triangle Park, No. Carolina 27706

1

Dr. M. J. Fischer  
Defense Communications Agency  
1860 Wiehle Ave.  
Reston, VA 22070

1

Dr. Richard Lau, Director  
Office of Naval Research Branch Office  
1030 East Green Street  
Pasadena, CA 91101

1

Dr. A. R. Laufer, Director  
Office of Naval Research Branch Office  
1030 Green Street  
Pasadena, CA 91101

1

Dr. James R. Maar  
National Security Agency  
Fort Meade, MD 29755

1

Miss B. S. Orleans  
Naval Sea Systems Command  
(Sea 03F)  
Rm 10S08  
Arlington, VA 20360

1

Mr. F. R. Priori  
Code 224  
Operational Test and ONRS  
Evaluation Force  
Norfolk, VA 20360

1

Dr. A. L. Slafkosky  
Scientific Advisor  
Commandant of the Marine Corps  
Washington, D.C. 20380

1

Mr. Glenn F. Stahly  
National Security Agency  
Fort Meade, MD 20755

1

	No. of Copies
Mr. David A. Swick Advanced Projects Group Code 8103 Naval Research Lab Washington, D.C. 20375	1
Dean of Research Code 012 Naval Postgraduate School Monterey, CA 93940	1
Naval Postgraduate School Monterey, Ca 93940	
Attn: M. J. Sovereign, Code 55	1
D. P. Gaver, Code 55	1
R. Stampfel, Code 55	1
J. M. Wozencraft, Code 74	
J. Yen, Code 55	10